MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 879 June 1986

Classifying Objects from Visual Information

Aaron Bobick
Whitman Richards

ABSTRACT: Consider a world of "objects." Our goal is to place these objects into categories that are useful to the observer using sensory data. One criterion for utility is that the categories allow the observer to infer the object's potential behaviors, which are often non-observable. Under what conditions can such useful categories be created? We propose a solution which requires 1) that modes or clusters of natural structures are present in the world, and, 2) that the physical properties of these structures are reflected in the sensory data used by the observer for classification. Given these two constraints, we explore the type of additional knowledge sufficient for the observer to generate an internal representation that makes explicit the natural modes. Finally we develop a formal expression of the object classification problem.

Acknowledgments. This report describes research done within the Psychology Department and the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. Support for this research is provided in part by the National Science Foundation grant ISI–8312240 and AFOSR F49620–83–C–0135. Support for the A.I. Laboratory's research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014–85–K–0124. We gratefully acknowledge the comments of Shimon Ullman, Steve Pinker, Dan Osherson, Patrick Winston, and especially Eric Saund.

1 Introduction

One of the major difficulties in designing a visual recognition system applicable to the natural world is that we do not have a single, predefined set of objects or models on which to map incoming image information. Unlike an assembly line where there may be only 14 possible objects, each having only a few stable configuration states, the natural world has an infinite variety of objects and viewing conditions. A system designed for recognizing trees will not naturally extend to recognizing fish. As such there is no simple set of target categories that spans all possible inputs. Yet clearly, we have no difficulty in recognizing a tree as categorically different from a fish. What set of categories does the observer use as a basis for recognition and how does the observer acquire this set of classes? This is the problem we address.

To facilitate the analysis of the problem we divide the issue into two parts. First, we propose the existence of a natural set of object categories as defined by the structure of the natural world; evidence is presented for this structure from both evolutionary biology and cognitive science. Second, given this claim, we address the issue of recovering these natural classes. It is demonstrated that the recovery of these classes is an under-constrained problem, requiring the observer to be given some additional information. The need for additional constraint is akin to other computational vision problems which require constraint to be embodied in the observer (e.g. structure-from-motion); we consider different forms of constraint that may permit the recovery of the natural classes. The goal of our research is to understand the type of information required by the observer to guarantee successful classification given different world classes and different criteria for success. A simple model and example is presented to illustrate some of the central issues, most important being how the natural mode constraints are embedded in the classification procedure. Then, the components of the object classification problem are defined formally. Finally, consideration is given to the development of a classification system designed to operate in the natural world.

2 The Goal of Recognition

Before we can design a set of categories for a visual recognition system, it is necessary to clearly define the goal of such a system. We propose that the first goal of a recognition system is to place objects into categories that are useful to the observer. We define a useful category as one which permits inferences about an object's potential behavior relative to the observer and his environment. That is, the observer uses sensory information to infer properties of an object that are important for the observer to know. These properties

may include function (*"that object looks like it belongs to the category of objects which are good to sit upon"*) or affordances [Gibson, 1979] (*"that object looks like one of those things which attacked me yesterday."*).[1] In some real sense this ability to make such inferences is the key role of any sensory system.

Given this first goal of a recognition system, is it possible for a naive observer to perform such a categorization of objects given only sensory data and no a priori knowledge about the objects he might encounter? Will his categorization permit the inference of potential properties or behaviors? The answers to these questions clearly depend on the domain in which the recognition system is to operate. If there is no correlation whatsoever between the sensory data and the behavior of an object, then no such inference is possible. For example, if every object in a world (including witches, bicycles, and trees) is spherical in shape, blue in color, and matte of surface, then such visual attributes would be useless for inferences important to the observer. Under such circumstances a visual recognition system which performed classification could not be built. Therefore, if we are to claim that the goal of the recognition system is to place objects in the world into useful categories, then it must be the case that the *world* is structured in such a way as to make these inferences possible. This is a strong claim, and one which is fundamentally different from stating that the only structure present is that which is imposed upon the world by the observer.

Stating that there is something special about the world which permits the the formation of useful visual categories suggests an approach for the design of suitable set of categories for visual recognition in the natural world. Specifically, let us consider what phenomena in the world cause it to exhibit the necessary properties which permit the inference of behavior from sensory data.

3 The Claim: Nature and Natural Modes

Nature's Categories

Consider the Gedanken experiment of giving a grade school art class the assignment of drawing pictures of imaginary animals — animals the children have never seen or about which nothing has been said. The results are as varied as the children who produce them: multiple-headed "monsters", flying elephants, and other composite animals are

[1] Notice we are not defining an object by its function as does Winston, etal. [1983]. Saying that *"all objects of category C have function F"* is quite different from saying that *"if an object has function F, then it is a member of class C."* Our categories are to be sensory categories (e.g. membership will depend on visual features) but will be formed such that they share common functional properties.

produced. Completely bizarre-looking creatures also emerge. There seems to be no limit to the number of animals that one could imagine, yet live only in the mind.

If these animals could exist, (i.e. if they could be built with biological hardware) why don't they? In some instances, the laws of physics simply preclude their feasibility. Flying elephants would require a weight, surface area, and muscle relation that cannot be created from the biological hardware used to make an elephant [McMahon, 1975]. Other animals, although feasible, may not exist because such creatures were either never formed by mutation, or, if formed, they were made extinct by forces in the environment. In this latter case and in the case of impossible animals, we can view the situation as an entity (the animal) which did not satisfy the environmental constraints in effect at the time. In fact, given the complexity of the natural world and the extensive pressures brought to bear by Nature on an organism, most arbitrarily-designed animals would perish, because the chance of creating arbitrary organisms which would be well-suited to the environment is almost zero.

As such, the existing species are special in an important way. They represent finely tuned structures capable of survival given the myriad of negative environmental pressures; they are Nature's solution to the constraint-satisfaction problem imposed by the environment. Survival of the fittest is simply a statement that the surviving species satisfies the environmental constraints better than any other species competing for the same resources.

For the purposes of our discussion, there are two aspects of Nature's solutions which are critical to perception (actually cognition in general). First, the solutions tend to be complex, and very broad in scope. By this we mean that there is no small set of properties of the organism which is sufficient for its survival. For example, fish have many properties in common to facilitate life in an aquatic habitat. Fins, streamlined contours, eyes capable of seeing in every direction from which a predator can attack — these attributes combined with a vast set of internal structures permit fish to survive.

The second important aspect of Nature's answer to environmental pressures is that the solutions tend to be disparate. That is, there does not appear to be a continuum of creatures each being capable of survival [Stebbins and Ayala, 1985]. The reason for this is clear. Let us consider two organisms, identical in almost every way except for some slight change in the second. Further let us assume that this variation is along a dimension which is ***significant to the creature's ability to survive.*** If the difference in capabilities is sufficient, one organism will be reliably superior to another organism, and if permitted to compete, that organism will be the victor: the "winner takes all." The pressure of natural selection moves the evolution of species to a discrete (or clustered) sampling along those dimensions relevant to the organisms' survival. The fact that nature is driven to a clustered distribution along the "important" dimensions, where important is defined to be relevant to an organisms interaction with its environment, is essential to our proposed solution to the problem of categorizing objects into useful classes. We refer

to the clustering of species as the "Principle of Natural Modes." In order to emphasize that this clustering is a claim made about the natural world, we label it as such and restate it as follows:

> **Claim 1a:** Environmental pressures force objects to have non-arbitrary configurations and properties that define object categories in the space of properties important to the interaction between objects and the environment. (Principle of Natural Modes)

In a few moments when we consider evidence for natural modes, we will discuss the validity of this claim for man-made objects, where the environmental pressures are obviously quite different.

Having made this claim, there are two important points which should be made. The first is that we are not stating that there exist objective categories in the world, *independent of any categorization criteria*. Rather, we are stating that there exists a clustering along dimensions which are important to the interaction between the object and its environment. Therefore, if some sensory apparatus is encoding properties related to these important dimensions, then there will be a clustering in the space defined by that sensory mechanism. The work of Rosch, etal. [1976] and Jolicoeur, etal. [1984] provide empirical evidence for the existence of visual categories from which useful properties can be inferred. Notice that the existence of mental categories does not imply the existence of categories in the world, only that the world is structured in such a way as to permit the formation of visual categories which are useful to observer. Therefore the ability to create such a categorization is a necessary condition for the expression of natural modes in observable properties.

The second point is that the Principle of Natural Modes is similar to Marr's "Fundamental Hypothesis" which argued that if a collection of certain observable properties tended to be grouped, then other properties (unobservable) would tend to group similarly [Marr, 1970]. The principal difference is that Marr did not provide a motivation for why one would expect to find certain observable properties grouped in clusters. In fact, claim **1a** by itself is not sufficient to provide a clustering of objects in the feature space of observable properties. Therefore we extend our claim with the following addition:

> **Claim 1b:** The properties which are important to to an object's interaction with its environment are (at least partially) reflected in observable properties.

Fortunately, **1b** is easily justified. For example, the basic shape of an object usually constrains how the object interacts with its environment. The legs of an animal permit it mobility. The color of an object is often related to it's survival: plants are green and polar bears are white. As such, the important aspects of an object tend to be reflected in properties which are observable. Therefore, claim **1b** taken together with claim **1a**

provide a basis for why one might expect to find a clustered distribution of objects in an observer's feature space.

Evidence for Natural Modes

Claiming the existence of natural modes is making a statement about the natural world. As such, evidence from the world should be available to support this claim. One such source of support comes from the field of evolutionary biology. Mayr (1984) states:

> [The biological species] concept stresses the fact that species consist of populations and that species have reality and an internal genetic cohesion owing to the historically evolved genetic program that is shared by all members of the species.

The objective existence of species represents a structuring of the world *independent of the observer*. Of course such a structuring is only useful if it coincides with the goals of the observer.

Man-made objects (actually most inanimate objects) are also subject to constraints upon form, although the environmental pressures are different. For example, a chair must have certain geometric properties to be able to function appropriately. It must allow access and stability, placing significant constraints on it's shape. A table must have a flat nearly horizontal surface with a stable support to function as a table. An even more complicated set of constraints related to ease of manufacturing and peoples' aesthetic interests operates on most constructed objects. Why is it that most books have similar aspect ratios? The common visual scene of "row houses" is an example of structure imposed by man mimicking the type of natural modes produced by nature. For a more extensive discussion about constraints on the shapes of objects and the non-arbitrary nature of objects see [Winston, etal., 1983; Lozano-Perez, 1985; Thompson, 1961]. Even chaotic processes may exhibit modes of behavior [Levi, 1986].

Utilizing the Natural Modes

Recall that our goal is to construct a set of visual categories onto which the observer is to map incoming image information; these categories must allow the observer to infer important properties about the objects. If we assume the existence of natural modes, we can make the following claim about the appropriate set of categories for visual recognition:

> **Claim 2:** If an observer is to make useful inferences about objects' behavior then he should categorize objects according to their natural modes.

This second claim follows naturally from our proposed goal of recognition and claim **1a**. Given that the observer is seeking to infer the properties which describe how an object interacts with it's environment, and given that these properties cluster according

to natural modes, then the observer should attempt to categorize objects according to their natural modes. Claim **1b** states that this goal can be accomplished using sensory data.

4 Object Processes and Natural Modes

Suppose one accepts the suggestion that natural modes exist in the world and as such are an appropriate set of target categories for a vision recognition system. How then does the observer acquire this target set? How does the observer recover the natural modes?

In order to illustrate and explore how one can perform object categorization using natural modes, we begin with a very simple world of objects generated by a language called LOGO, an educational tool developed at MIT to teach mathematics using graphics. We choose this domain because it has a formal structure with properties that are well known (Adelson and diSessa, 1984). A point on a screen is viewed as a little creature (turtle) that exists in a plane and responds to a few simple commands: FORWARD moves the turtle in the direction it is facing by some number of units. RIGHT rotates it in place clockwise some number of degrees. BACK and LEFT cause opposite movements. Beginning with this very simple vocabulary, a wide diversity of 2D patterns can be generated. Figure 1 shows some simple example "objects."

Let us cast the classification problem in terms of these LOGO figures. Upon inspection of the "objects" in Figure 1, a first impression is that there are two groups: one consisting of simple regular polygons (triangle, pentagon, octagon) and another of star-like objects where the lines intersect one another. (In fact, we will see shortly that there are three distinct groups, as defined by the behavior of the generating program, POLY.) The question we are investigating is what are the underlying principles used by the observer to classify these objects. The existence of some principles is demonstrated by the fact that most people see the same groupings. What information do we use and then how can we be sure that this information will allow us to converge to the "correct" classes? From our guesses about the classification, it is clear that such properties as the size of angle, the length of edges, the existence of intersections, etc., are possible features that guided our choices. But why these and not length to area, number of intersections divided by length or some other "weird" measure? In fact, we could group these objects solely upon the number of vertices. Those having less than five vertices form one group, those with five or more another. What then dictates the popular groupings of $A\,C\,E$ and $B\,F\,G\,D\,H\,J$?

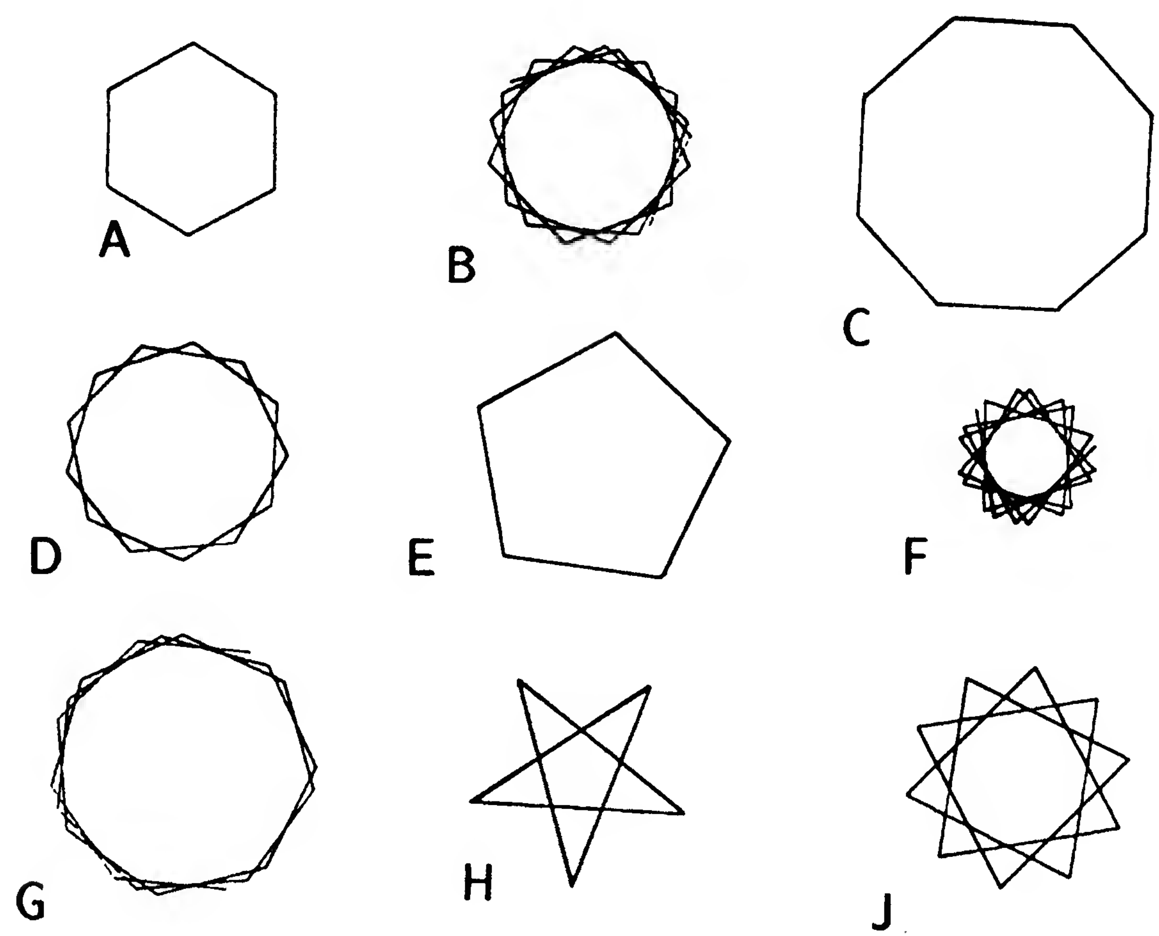

Figure 1. Some simple LOGO objects.

4.1 Object Processes and Properties

As discussed in section 3, we claim that natural modes occur because of interactions between processes that create objects and the surrounding environment. As a model of this interaction we propose a construct called an *object process*. An object process is a two component construct. The first part is a *generating algorithm* representing the mechanism responsible for creating the object; the second part, a *parameter rule* represents constraints acting upon the generating mechanism. In Fig. 1, we actually used three different object processes used to produce the figures. One of them is that which created objects $A\,C\,E$. (The fact that the distinctions between the remaining two processes is less immediately apparent is an interesting point to which we will return later.)

Generating Algorithms

We define a generating algorithm, G_k, as a procedure which given some input parameter, $\mathcal{A}$, produces as output some object θ. G_k may be thought of as a map from some set I_k, the set of all defined inputs to G_k, to the set of objects. Thus $G_k(\mathcal{A}_0)$ is the object produced by G_k for some specific input parameter $\mathcal{A}_0$. Shortly we will consider placing restrictions on the input parameter $\mathcal{A}$.

From our LOGO domain, the procedure POLY, used to generate the objects in Fig. 1, is an example of a generating algorithm. Thus all three object processes in Fig. 1 are common in the first component, namely the generating algorithm. POLY is defined as follows:

```
DEFINE POLY (S,A) ;  generating algorithm ''POLY''
    FOR S
    RIGHT A
    POLY (S,A).
```

For the moment, the values of the parameters S and A are chosen randomly: S is a length chosen from the positive real numbers and A is an angle ranging from 1 to 179° (0 and 180° are degenerate choices). The way POLY works is as follows: Angle A and side length S are chosen. The turtle then moves forward S units from its initial position. It then halts and turns right A degrees. The procedure is repeated with this new heading. As is evident from figure 1, different values of S and A produce several different types of shapes. A priori, we have no reason to believe that structures should emerge from running POLY. But they do. We will examine the structure that POLY imposes on its objects more closely.

Obviously, depending upon the generating algorithm G_k, the choice of the input parameter $\mathcal{A}$ may greatly affect the types of objects produced. Restrictions placed upon these input parameters constitute the second component of the object process and will be discussed in the next section. However, there may also be properties that are true of all objects produced by a given generating algorithm G_k, regardless of the input parameter. For example, all objects in Fig. 1 have the property that all the vertices of the objects will lie on a circle. Similarly, for each object there is an inner circle which is tangent to every line segment. We refer to these properties as *emergent properties*, those which are true because of the generating algorithm used to produce the objects. Thus if two different object processes have different emergent properties they must differ in their generating algorithms.[2]

[2]The word *property* is often a source of confusion. For example is "color" a property that takes on values of red or blue, or is "color is red" a property that is either true or false? At this point we will use the term loosely, with the meaning being clear from context. Later, we will

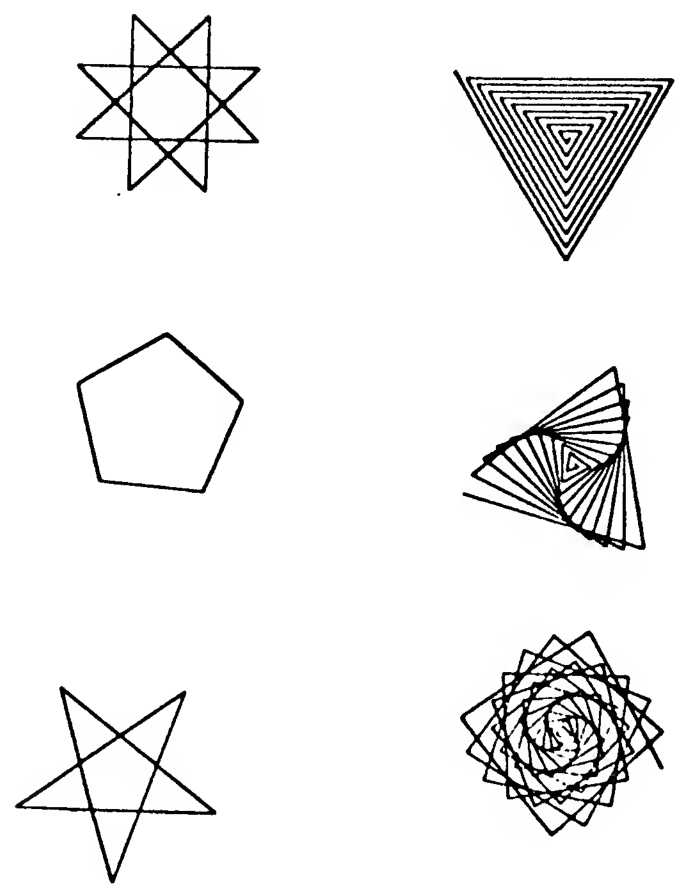

Figure 2. Objects generated by POLY compared with objects generated by POLY-SPIRAL.

Figure 2 illustrates the relationship between generating algorithms and the entailing differences in their emergent properties. The objects on the left were produced by POLY for different choices of S and A. The objects on the right however were generated by POLY-SPIRAL, a generating algorithm identical to POLY except that with each recursive call the length of the side is incremented by one:

```
DEFINE POLY-SPIRAL (S,A) ; generating algorithm ''POLY-SPIRAL''
    FOR S
    RIGHT A
    POLY (S+1,A).
```

Notice there is no inner circle tangent to all edges of the objects produced by POLY-SPIRAL, nor an outer circle on which the vertices lie. In fact the objects produced

use the term "feature" to mean a function measured on an object, and "property" as being a feature taking on a particular value.

by POLY-SPIRAL would not even be bounded if the algorithm were permitted to run indefinitely.

We should note that it is possible for a generating algorithm not to have any interesting emergent properties at all.[3] If this were true, then the particular algorithm used to produce the objects would be irrelevant. However, if we restrict ourselves to those generating algorithms which do have emergent properties, we have our first relation between object processes and natural modes: *Emergent properties represent structure shared by all objects created by processes with the same generating algorithm.*

Of course, for a given generating algorithm G_k, there will be properties of the objects produced by the algorithm which depend critically on the choice of input parameter $\mathcal{A}$. Referring again to POLY, consider the property of closure. Let us assume we allow POLY to make N recursive calls. Then the figure produced will be closed (will return to the starting point) if and only there exist two irreducible integers p and q such that $\mathtt{A} = 360\,p/q$ and $q \leq N$. That is for only some subset of the possible input parameters is the object closed. We therefore refer to properties such as closure for POLY as *parametric properties* — those properties whose value depends upon the input parameters.

[Because parametric properties of a generating algorithm are a function of the input parameter, we can qualitatively describe the relation between the selection of input parameter and the occurrence of the parametric property. For example, if we assume that the angle A of POLY is chosen randomly from a uniform distribution of the real (or rational) numbers between 0 and 180°, then for any finite number of iterations N, the likelihood of the figure being closed is zero. Thus we can consider closure to be a *non-generic* property for the generating algorithm POLY. By comparison, the property "having intersection" requires that A not be an integer divisor of 360°. Therefore "having intersections" may be described as being *generic* under POLY given the uniform distribution of A. Other types of property descriptions (e.g. stability) are possible if one is willing to assign a probability density to the input parameter range. Although we will not utilize these types of descriptions currently, we note the possibility of using them in the design of a classifier.]

Parameter Rules

The generating algorithm, the first component of an object process, constrains certain properties of the objects produced by the process. First, the emergent properties of the objects are fixed, regardless of the input parameter. Second, a mapping is established by the generating algorithm between possible input parameters and the properties of the output objects. As yet though, we have no constraint on the parameters selected. In POLY if the values of S and A are unconstrained, a large variety of objects may be produced.

[3]By *interesting* we mean that the property will have more than one value across the set of all outputs of all generating algorithms.

If we consider object processes to represent the interaction between procedures for generating the object and its environmental factors then we do not want all possible input parameters to be permitted for a given generating algorithm. Restrictions must be placed on the input parameters if the objects are to have non-arbitrary configurations and properties. This is our Principle of Natural Modes. In particular we might choose only those parameters that ensure an object's greatest chance for survival. If we assume that parametric properties are subject to environmental constraint, for example, then we need to restrict the input parameter $\mathcal{A}$. As such we introduce a ***parameter rule***, R_{ki} as the second component of an object process: A parameter rule, R_{ki} as applied to a generating algorithm G_k restricts the input parameter $\mathcal{A}$ to a subset of the possible inputs I_k. Any well defined set theoretic statement expressed in terms of subsets of I_k which evaluates to a set which is itself a subset of I_k is a valid parameter rule as applied to G_k. From POLY examples of parameter rules are *"A equals 45°"* and *"the quantity $D/sin(A/2) = 30$"*. Each of these restricts the input parameter to some subset of the possible inputs. To be complete we include the "null rule" which places no restrictions on $\mathcal{A}$.

To summarize, we have defined an object process to be a pair $\langle G_k, R_{ki}\rangle$ – a generating algorithm, G_k, whose inputs have been restricted to some subset of possible inputs by a rule, R_{ki}. In Fig. 1, one process produced the objects $A\,C\,E$. The generating algorithm was POLY; the rule, *"A equals $360/q$ where q is an integer"*. Of course to see these figures as a single object process requires a classification scheme utilized by the observer which is sensitive to whatever structure is present in those figures relative to the population. That is, the observer must be implementing some method of classification that indicates why the figures $A\,C\,E$ belong together. Notice that any arbitrary classification might be produced; without additional information all groupings are equally valid. Indeed there are over 20,000 possible categorizations of these nine objects. Therefore the observer must be making use of some additional constraint to make the judgment as to what the appropriate partition is. In the next section we will develop the concept of classes and class constraints, a modeling of the what the natural modes look like as expressed in terms of object processes.

4.2 Natural Modes and Classes

In section 3 we made the claim that the goal of a visual classification system should be to classify objects according to natural modes. Using object processes as a model of the interaction between generating algorithms and the environment, which give rise to the natural modes, the goal becomes to group objects according to the object processes that created them. As such we will define a *class* Θ_η as all objects that can be produced by some object process $\mathbf{OP}_\eta \leftrightarrow \langle G_k, R_{ik}\rangle_\eta$. That is, the notation $\mathbf{OP}_\eta$ refers to an actual object process, and Θ_η is a set of objects $\{\theta_1, \theta_2, \ldots\}_\eta$ which could be produced by $\mathbf{OP}_\eta$. In the objects of Fig. 1, the regular polygons $A\,C\,E$ form a class. (One could not know

this fact without some additional knowledge either about POLY or about the generating algorithms existent in the LOGO world.) Clearly a class may contain an infinite number of objects, even though at any given time the observer has only viewed a finite number of objects. We can therefore think of the object process itself as a representation for the class.

We can now specify our general problem more precisely. We consider a *population* of objects to be the objects actually created by various object processes, and therefore a subset of the union of one or more classes. Our goal then can be stated as: *Given a population of objects in the world Θ_W, where Θ_W is a subset of the union of one or more classes, $\Theta_W \subseteq \bigcup_\eta \Theta_\eta$, partition Θ_W into subsets $\widehat{\Theta}_\eta$ such that $\widehat{\Theta}_\eta = \Theta_W \cap \Theta_\eta$. That is recover each class present in the population.*[4] Notice that we intend the term "partition" to convey its mathematical definition of disjoint subsets, implying that each particular instance of an object belongs to only one class. That is any single object present was created by only one generating process. This does *not* imply that two classes must be disjoint; as yet we have placed no constraint on the relation between one class and another. Rather, we simply state that classification should produce disjoint subsets of the given population and that these subsets should be in a 1–1 relationship with the actual classes present. Is should be noted that we have only required the observer to recover the class structure in Θ_W, the viewed population of objects, as opposed to recovering the entire classes Θ_η which includes objects not yet seen. Later, when we give a formal definition of the classification problem, we will extend the goal of classification, requiring not only the recovery of the classes present in the viewed population, but also the recovery of the actual classes present in the world.

Given this goal of classification according to object process, the appropriate question to raise is under what conditions can this goal be achieved? For example, if two different object processes are capable of producing objects with identical observable properties, then clearly the classification goal is unattainable.[5] Therefore we make use of claim **1b** to clarify the relation between our domain of objects and object processes: The differentiating properties of the object processes $\langle G_k, R_{ki}\rangle$ will be reflected in differences between the visual (sensory) descriptions of the objects $\{\theta\}$. Note that if $G_k \neq G_j$ then, assuming that the generating algorithms have some different emergent properties, there will automatically be a set of differentiating properties.

Thus, embedded in the visual properties of objects we assume there is evidence of the different object processes present. But how do we find this evidence? Which of the

[4] We use the circumflex $\widehat{\Theta}$ to represent proposed classes by the observer. This notation is motivated by parameter estimation theory which represents estimates of actual parameters by circumflex.

[5] In the natural world, whales and fish are examples of significantly different object processes creating objects with almost identical visual properties. If this type of deception were the rule rather than the exception, visual classification would impossible in general.

many observable properties are relevant to the classification? For example in Fig. 1, there are numerous ways to partition the nine LOGO objects: even vs. odd number of vertices; number of sides greater or less than 10. Given that there are over 20,000 ways to partition the nine objects into disjoint subsets, which do we pick?. Without additional information there is no rationale for preferring one partition over another.

Obviously, we require constraints on the types of object processes possible as well as constraint on the relations between object processes. These constraints should reflect the structure present in the world. For example, suppose the observer "knew" that in the LOGO world "boundedness" is determined by the generating algorithms used to produce objects; that is, the observer will *assume* that boundedness is an emergent property. Then the observer can immediately deduce that in Fig. 2 there are at least two classes of objects present, namely those produced by POLY and those produced by POLY-SPIRAL. Because "boundedness" is an emergent property it constrains all objects of a given class. Such *class constraints* force objects into categories that are observable. Thus, we want the class constraints to encode the notion of Natural Modes, causing the observer to recover the actual classes present. Without such constraints, the observer cannot infer that a particular classification is correct.

Given that our goal is to group objects according to object processes, and given that we have identified a relation between object process and natural modes, we need to develop formal definitions of class constraints that reflect the structure found in objects because of the presence of natural modes. We have already given one clear example: restricting a property to be emergent makes that property relevant to classification. However, we have not specified the structure of natural modes explicitly enough to translate that structure into additional class constraints. Nor have we specified how the class constraints are to be embedded in the classification process. To gain some further insight into the natural modes concept, as well as to introduce classification procedures, we will proceed to categorize the POLY objects of Fig. 1. Later we will discuss more formally the issues raised.

5 An Example Classification — POLY

Our objective is to classify the objects of Figure 1 generated by POLY. The goal of this section will be to demonstrate that the observer can successfully recover the real classes in the world only if the constraints embodied by the classification procedure accurately reflect the structure of the world. Our criteria for success will be that the classification procedure groups the objects according to the different object process used to create the

objects. At the outset, the observer has no knowledge of these processes, nor how many classes are present.

5.1 Data: The Real Classes

The "real" classes are subsets of the population which are generated by different object processes. Formulating the problem this way allows us to set up competence and performance criteria. We "know" what the true classes are and, as such, what the correct classification looks like. Therefore we can investigate questions about the conditions that will enable the observer to achieve the correct classification, how quickly he will converge to a stable configuration, and whether the strategy lead to a null solution, where the constraints can no longer be satisfied.

The objects in Figure 1 were generated by POLY according to three different rules. As such there are three true classes; each class is associated with the generating algorithm of POLY plus a different parameter rule. As we will see, all three classes can be recovered only when the class constraints correctly describe the relationships between these rules. The parameter rules are as follows:

R_1 : Angle A equals $360°/q$ where q is an integer less than N, the maximum number of iterations of POLY. S is random uniform from 1 to 10.

R_2 : Angle A is uniform random from 1° to 179°. S equals 15.

R_3 : Angle A is $360° \cdot p/q$ where p and q are integers and q is less than N. $\mathtt{S}/\sin(\frac{A}{2}) = 30$.

Each of the above rules constrains some of the properties of objects generated by POLY. However, the degree to which these rules effect our representation of the objects depends upon the properties used to describe the objects.

5.2 Properties, Features, and Values

Although the object processes are responsible for creating the objects present, our classification will be based upon our descriptions of the objects. Therefore we need a ***representation*** that is is computed for each object and that is used as the basis for classification. Throughout this paper we have referred to visual ***properties*** without providing a formal definition. Let us define a ***feature*** as a function which takes as its argument an object and returns some ***value***. For example, "length" would be a feature and "20" would be a value. In this notation, a visual property can be defined as a particular feature having a particular value. As such, "length of 10" and "having the color blue" would be a visual properties. We assume that the observer is able to recover the values of these

features from the sensory data. In order to begin to proceed toward a solution to the visual classification problem, we will initially assume that a sufficient set of features is available a priori to the observer. That is, the observer has a priori knowledge of the set of possible features on which successful classification can be based. At first glance, one might expect that with such a major assumption, the classification problem becomes trivial. Our example (POLY) will show that this is not the case at all.

To work through the example, six features have been chosen to represent the objects. Included in this set are features relevant to classification, as well as those which are irrelevant (e.g. angle). Table 1 provides a list of the features along with the values they can take; the behavior of these features and their values under POLY (expressed in the terms introduced in section 3) is also stated. Given these features, we measure their values for the objects in Fig. 1 (Table 2). These valued features will be the basis for classification; these features will be all the information known about the objects. Finally, we can describe what each "real class' would look like in terms of these features. Table 3 shows the range of feature values that each class will have. Notice that contained in the table are both emergent and parametric feature values.

Feature	Values	Comments
Closure	$\{t, f\}$	Parametric, f is generic.
Vertices on Circle	$\{t, f\}$	t is emergent.
Has Intersections	$\{t, f\}$	Parametric, t is generic.
Side length (nearest integer)	$\{1, 2, \ldots, 10+\}$	Parametric.
Diameter of Bounding Circle (nearest integer)	$\{1, 2, \ldots, 100+\}$	Parametric.
Angle	$\{1, 2, \ldots, 179\}$	Parametric.

Table 1. Features used to describe objects in POLY.

The above description represents the "true" state of the world, the data available to the observer (Table 2). The goal of the observer is to recover the classes shown in Table 3.

5.3 Example Class Constraints

To find the classes present in the population displayed in Figure 1 we must impose further constraints that capture the Principle of Natural Modes. Without such constraint, all the objects could have been generated by the process $<$ POLY, $R_\emptyset$ $>$ where $R_\emptyset$ is *"S is random uniform* $(0, 10^{120}]$*, and A is random uniform* $(0°, 180°)$*"*. Alternatively $R_\emptyset$ could

Object	*Closure*	*V. on C.*	*Intersections*	*Side Length*	*Diameter*	*Angle*
A	*t*	*t*	*f*	13	26	60
B	*f*	*t*	*t*	15	27	68
C	*t*	*t*	*f*	16	42	45
D	*t*	*t*	*f*	14	30	55
E	*t*	*t*	*f*	21	36	72
F	*f*	*t*	*t*	15	18	110
G	*f*	*t*	*t*	15	35	50
H	*t*	*t*	*t*	29	30	144
J	*t*	*t*	*t*	24	30	108

Table 2. Feature values for the objects of Figure 1.

Feature	Class 1	Class 2	Class 3
Closure	*t*	*f*	*t*
V. on C.	*t*	*t*	*t*
Intersections	*f*	*t*	*t*
Side Length	$\{1,2,\ldots,10+\}$	15	$\{1,2,\ldots,10+\}$
Dia. Bounding Circle	$\{1,2,\ldots,100+\}$	$\{15,16,\ldots,100+\}$	30
Angle	$\{1,2,\ldots,179+\}$	$\{1,2,\ldots,179+\}$	$\{1,2,\ldots,179+\}$
Objects	A, C, E	B, G, F	D, H, J

Table 3. Feature descriptions for the classes showing the range of values possible for objects from each class.

be a parameter rule which only allows the nine different possible combinations of S and A required to produce the nine objects in Fig. 1.

We will provide the extra constraint in terms of class constraints which reflect the claim that objects are clustered in natural modes in the world. Recall that such constraints are going to be used by the observer to achieve a classification. An example of a class constraint would be that any object class will have at least two properties in common. That is, at least two features will be fixed to some value. It is the hope of the observer that such constraints match well the constraints in the world which generated the "real classes." If so, his classification will be driven to matching the actual generating processes; if not, his classification procedure will be unsuccessful.

For the example of POLY figures, we will use several simple class constraints. Some of the constraints will exactly reflect the structure of the generating processes, whereas

others will be approximations. An important question to investigate is how does the behavior of the classification mechanism vary as the constraints vary in their accuracy.

Our first type of class constraint is a restriction on the structure of any individual class:

> **CC1a:** Any class will have at least 2 properties fixed,

> **CC1b:** Any class will have at least 3 properties fixed,

These constraints are examples of ***intra-class*** constraints. They act upon a class independent of the other classes present. Two versions are presented to be able to compare the success of a classifier depending upon how well the class constraints match the world.

A second constraint type is ***inter-class***. For the POLY example the following constraints are of this type:

> **CC2a:** Any two classes will differ by at least 1 fixed property.

> **CC2b:** Any two classes will differ by at least 2 fixed properties.

Such constraints operate across classes, restricting the structure that the set of classes may exhibit. Therefore, combined with the intra-class constraints above, we are able to specify the overall structure that a classification should attain. We believe this is essential (as do all proponents of any cluster analysis techniques) in discovering the important structure (i.e. the classes) in the data.

The last type of class constraint is quite different from the first two. It is a constraint on the types of properties which are constrained by object processes. Our example is:

> **CC3a:** Object processes fix at least 2 *parametric* properties.

> **CC3b:** Object processes fix at least 3 *parametric* properties.

This is an important type of class constraint in that it relates classification criteria to types of properties, which are functions of the object process. In this case, a class is required to fix the values of features that depend upon the input parameter of the generating algorithm. As such, restrictions have been placed on what types of properties may be used to define a class. Such constraints reduce the amount of information that needs to be considered when attempting to recover the classes, and are therefore helpful in controlling the the classification process. Also, and perhaps more importantly, this type of constraint can restrict the creation of new feature, since the system may be able to know, ***a priori***, whether or not a feature is likely to be of one type or another. To do this would require knowledge of the types of generating processes that occur in the world, but perhaps knowledge of this type is not unreasonable. Again, we will discuss this further when considering the natural world.

Notice that to be able to exploit a constraint like **CC3** the observer must be able to know that a feature is parametric (or non-emergent). Therefore, the observer must also be provided with a method a making such a determination. For the example of POLY, we will have the observer assume that anything that is true about the entire population is emergent, caused by the generating algorithms used to produce the objects. Alternatively, the observer could simply be told that a particular property is emergent: e.g. "boundedness" is an emergent property.

Using the above three types of constraints and some logical combinations of them we wish to recover the classes in Fig. 1. To do this task requires a classification procedure or method that the observer will execute.

5.4 Example Methods

Given a population of objects, a set of properties to describe the objects, and class constraints representing the structure to be reflected in the classes, how does one procedurally determine the real classes? We term such a procedure or algorithm a *classification method.* A method is a procedure by which the observer generates classifications which satisfy the given class constraints.

As a simple example consider the following naive method to discover the classes present in a population: Given n objects, generate all possible partitions of the objects and select the partition which satisfies the class constraints best. This is an example of a method which is guaranteed to find the best class description, but at a tremendous, in fact unmanageable, cost [Rota, 1964]. The problem lies in the fact that the number of possible partitions of a set with n elements is $\sum_{k=1}^{n} S(n,k)$ where $S(n,k)$ is the Stirling formula for the number of partitions of a set with n elements into k disjoint subsets. Stirling's formula is computed recursively, being defined as $S(n,k) = k\,S(n-1,k)+ S(n-1,k-1)$. For example, $S(15,3) \approx 2 \cdot 10^6$: there are over two million ways to partition a set of 15 objects into just 3 groups! Given only 15 objects there are $\approx 1.4 \cdot 10^9$ possible partitions: over a billion possible candidates for the class grouping. Obviously the combinatorics of such a method are prohibitive.

For the POLY example, we will use a simple incremental method. This method tells the observer what to do as each new object is seen; the classification is achieved incrementally as objects are encountered. Our method is:

> **M1:** If the new object can be added to an existing class while still satisfying the class constraints, do so. If the object can be added to more than one class, chose arbitrarily. If the object cannot be included in any existing class, form a new class containing only that object.

The basic approach is to include objects in existing classes if at all possible. This method is is similar to a procedure called *divisive clustering* in the cluster analysis literature [Duda

and Hart,1973].[6] This type of strategy significantly simplifies the control problem by eliminating the computationally difficult operations of splitting and merging classes; the combinatorics of the procedure are greatly diminished. Of course, we have also sacrificed much power, as well as possibly making the scheme sensitive to the order of presentation of the objects. If our classification procedure is going to be effective, it will need to be the case that the method used is sufficiently powerful to eventually discover the correct classification. Therefore the design of the method must also reflect the constraints operating in the world.

5.5 Example Scenarios

Finally, we have all the necessary tools to work through the POLY example. We will consider four different scenarios, using different combinations of class constraints and object sequences. The first scenario will be discussed in detail, introducing the method of class generation. The following examples will be highlighted where interesting events occur. For feature based descriptions of the objects refer to Table 2.

Under-Constrained Case

The first scenario uses method **M1**: add the object to an existing class if possible, and the class constraints, **CC1a** and **CC2b** which require that each class have at least two fixed properties, and that different classes will differ by at least 2 fixed properties. We proceed one object at a time.

The first object seen is A, which obviously cannot belong to any previous class; it therefore belongs in its own category. Object B is seen next. Since A and B do not share any two features, they can not be in the same class according to **CC1a**. As such it becomes it's own class as well. Object E follows, and since it shares several features with A and only one (which is actually an emergent property) with B, it is grouped with A. Notice that we now have reduced the possible fixed features of the first class. Since A and E differ in the length of the side and the bounding diameter, then those are not candidates for the fixed features of the class.

The observation of object H causes the classifier to be in a non-determined position. If H is added to the A,E group, then the first class is defined as having $\{closure = t\}$ and $\{Vertices\,on\,Circle = t\}$, with the fixed properties of the second class yet to be determined. If added to the class with B, H causes the A,C class to be defined by $\{closure = t\}$ and $\{Intersection = f\}$ and the B,D class defined by $\{Vertices\,on\,Circle = t\}$ and

[6] *Cluster Analysis* refers to methods developed by pattern recognition theorists to help identify structure in their data. As such it bears a strong resemblance to the problem of classification presented here. In section 6 we briefly contrast our formulation of the visual categorization problem against standard clustering techniques.

Scenario 1: Under-constrained case.

Method: **M1** – incremental with arbitrary decision

Constraints: **CC1a** – Fix 2 properties.

CC2b – Classes differ by 2 properties.

Object	Class	Classification				Comments
A	1	A				Start
B	2	A	B			No two in common.
E	1	A,E	B			
H	3	A,E,H	B	A,E	B,H	Arbitrary choice.
F	2	A,E,H	B,F	A,E	B,H,F	
C	1					
D	3	1,2	3	1	2,3	Converged.
G	2					

Scenario 2: Correctly constrained case.

Method: **M1** – incremental with arbitrary decision

Constraints: **CC3a** – Fix 2 *parametric* properties.

CC2a – Classes differ by 1 fixed property.

Object	Class	Classification			Comments
A	1	A			
B	2	A	B		
E	1	A,E	B		
H	3	A,E	B	H	H doesn't share with any two
F	2	A,E	B,F	H	
C	1	A,E,C	B,F	H	
D	3	A,E,C	B,F	H,D	Converged to correct description.
G	2				
		1	2	3	

$\{Intersection = t\}$. Either state is stable: the additional objects will not change the classes. Since the method **M1** instructs the observer to choose arbitrarily when necessary let us assume that he does so. At this point the "die is cast" and the classification procedure will simply continue to add all new objects to one of the two classes present.

The classification method has converged; but, were we successful? We have partitioned the population into two groups, and we have partially succeeded in recovering the "real classes." All objects from any real class are all in one of the created classes. However, we have not recovered all the structure that distinguishes between the classes. First, we were required to make an *arbitrary* choice for H, not one driven by the data.

Second, regardless of the choice of class for object H we do not recover all three classes.

What happened? The principal problem was the class constraints were too weak. The constraints permitted a description of the observed data which was less dissociated than the real classes. Since the method being employed was that of starting from the least dissociated description and changing to a more dissected classification only when forced to by the class constraints, this coarser partitioning was the classification found. This is an example of the constraints used by the observer (the classifier) not being correctly matched to the constraints imposed upon the generating processes which created the actual classes, as well as a demonstration between the interaction between the accuracy of the class constraints and the type of classification method used.

Correctly Constrained Case

Let us remedy the situation. In scenario 2, the constraint **CC2b** is replaced by **CC2a**, changing the number of differing fixed properties to 1. Also, constraint **CC1a** is replaced by **CC3a**: two properties fixed by the generating processes will be parametric. To implement this constraint, we will assume that the observer is allowed to look at a large sample of the population in a "look ahead" manner; this is adding a small degree of non-incremental behavior to the procedure. If all the samples are found to have one particular feature value, the observer will assume that the value is emergent, not parametric, and thus will not be counted as one of the 2 fixed parametric properties.

Stepping through the scenario, we keep the sequence of objects the same as in scenario 1. The first three objects A,B,E are grouped as before. However, when object H is encountered the situation changes. If H is added to either of the existing classes, the intersection of the values of the objects in that class would require that $\{Vertices\,on\,Circle = t\}$ be one of the fixed values of the class. But according to **CC3a**, that feature specification would not be permitted, since all the objects in the population have $\{Vertices\,on\,Circle = t\}$ making t look like an emergent value for the feature $\{Vertices\,on\,Circle\}$. Therefore, object H would become a class of it's own. Now, the remainder of the objects would be grouped accordingly as the observer has converged to the final classes. Additional objects will serve only to refine the description of the classes — determining the features fixed by each generating process. This scenario is a demonstration of a classifier exhibiting *performance*: solving the task of recovering the natural classes on this particular sequence of objects.

In fact, this example helps us to refine our notion of what it means for an observer to be successful. Not only does the classification method partition the objects viewed according to the actual classes present, but the description of the classes remains constant

Scenario 3: Second correctly constrained case, new order.

Method: **M1** – Incremental with arbitrary decision

Constraints: **CC3a** – Fix 2 *parametric* properties.
CC2a – Classes differ by 1 fixed property.

Object	Class	Classification			Comments
J	3	J			
E	1	J	E		
C	1	J	E,C		
G	2	J	E,C	G	G doesn't share with any two
A	1	J	E,C,A	G	Converged to correct description.
D	3	J,D	E,C,A	G	
B	2	J,D	E,C,A	G,B	
F	2	J,D	E,C,A	G,B,F	
H	3	J,D,H	E,C,A	G,B,F	
		3	1	2	Actual classes.

Scenario 4: Over constrained case. No convergence.

Method: **M1** – Incremental with arbitrary decision

Constraints: **CC3b** – Fix 3 *parametric* properties.
CC2b – Classes differ by 2 fixed properties.

Object	Class	Classification						See text for Comments
A	1	A						
B	2	A	B					
E	1	A	B	E				
H	3	A	B	E	H			
F	2	A	B,F	E	H			
D	3	A	B,F	E	H	D		
C	1	A	B,F	E	H	D	C	
G	2	A	B,F,G	E	H	D	C	
J	2	A	B,F,G	E	H	D	C	J

as more objects are seen. Later when we formalize our description of the classification problem we will define successful classification to consider this type of convergent behavior.

Scenario 2 demonstrated the power of class constraints matched correctly to the real generating processes. Correct constraints make the recovery of the natural classes possible. However, was it simply fortuitous that we were guided to the correct classification

by the objects present, and is it possible that with another sequence of objects we would have failed to arrive at this solution? In scenario 3, we use the same constraints and method as scenario 2, but a different sequence of objects. Yet the classifier converges to the same classes as before. In fact for this set of constraints and this method one can show that for *all* sequences of objects of drawn from the population generated by the rules of the example, the classifier will converge to the same set of classes. As such, we can state that the classifier given class constraints **CC2a** and **CC3a**, and method **M1** has the *competence* to recover the natural classes in a population constructed like that of Fig. 1.[7] This is an important point since the goal of this paper is to study the amount of knowledge a classification system must be given *a priori* to be able to have the competence to recover the real classes present in constrained populations.

Over-Constrained Case

Finally, as a last example, we consider the over-constrained case; it may be thought of as the extreme opposite of the non-deterministic case. In Scenario 4, the constraints in effect are **CC2b** and **CC3b**; notice that these constraints are do not correctly describe the world. From table 2, one can see that class 1 does not fix 3 parametric properties. Also classes 2 and 3, though having 3 parametric properties fixed, do not differ by two fixed properties. Stepping through the scenario we see that the class constraints force the observer to propose many more classes than are present. Using the sequence of scenarios 2 and 3, we see that A and B are again separated immediately. However, when E is seen, although it agrees with object A in three properties ($\{Closure = t\}$, $\{Vertices\,on\,Circle = t\}$, $\{Intersections = f\}$) the $\{Vertices\,on\,Circle\}$ is not parametric. Therefore E is placed in its own category. When H is viewed, it cannot be added to any of the existing categories according to **CC3b**, and therefore also is made a solitary class. Object F is placed with object B since they share 3 fixed properties that are all parametric. However, now when object D is seen, it cannot be placed with object H (as it should if we were recovering the real classes) even though they share 3 parametric properties. This is because if D,H are a category, then their fixed properties must be $\{Closure = t\}$, $\{Intersections = t\}$, and $\{Bounding\,Diameter = 30\}$, which overlaps

[7] To show that the constraints **CC2a** and **CC3a** along with method **M1** are competent to recover the correct classes regardless of the sequence, consider each object as it is viewed. For each object θ_i in the sequence, either there currently exists a proposed class containing previously seen members of the same real class, or there does not. If there does exist such a proposed class, the new object will be added to it since each parameter rule guarantees the fixing of two parametric properties, satisfying **CC3a**. Therefore there will be *at most* one proposed class corresponding to each real class. If there does not exist such a class, the new object will become its own proposed class, since the parameter rules prevent objects from different classes from sharing more than one parametric property, requiring a new class according to **CC3a**. Thus there will be *at least* one proposed class for each new class. Therefore, the three classes will be recovered exactly.

with the fixed properties of the B,F category. Thus, objects which should be placed in similar categories will instead form new categories. The system will not converge to some stable set of classes.

Non-convergence is one example of the behavior of an over-constrained system. Alternatively, it can be the case that the classification method cannot generate a partitioning that is consistent with the class constraints. If the observer is not going to give up, he needs to be able to do more than decide to what classes objects belong. In particular, he needs to be able to either *i)* modify the class constraints, to make them compatible with the data as described by the current features, *ii)* change the feature list so that the data are consistent with the constraints, or *iii)* both. One of the more interesting consequences of the classification methodology outlined here is its extension to the learning of proper descriptions of objects by exploring constraint satisfaction.

6 Formalizing the Components

Having informally worked through the example of POLY objects, we will now formalize the important components of the classification problem. This task can be divided into two parts. First we need to describe in detail the model of the world employed by the classification procedure. Second, we need to define the components built into the classifier — the information embedded in the observer that is exploited when attempting classification. Some of the notation has been introduced in previous sections, but we shall review these terms for clarity. Several of aspects of the formulation of the vision classification problem have been influenced by the formal learning theory developed by Osherson, Stob, and Weinstein [1986].

6.1 Modeling the World

The world model introduced in sections 3 and 4 contained objects created by ***object processes***. The first component of object processes was a generating algorithm, G_k, defined over some input parameter I_k. For any input parameter $\mathcal{A} \in I_k$, G_k produces one object. A very simple yet non-trivial consequence of this type of model of object formation is that every object present in a population requires some generating algorithm to produce it. Therefore a hypothesis of a particular generating process to explain the existence of a particular object may have implications for the other objects seen by the observer. We will return to this issue when we consider the formalization of the class constraints and of the hypotheses proposed by the observer.

The second component of object processes was a parameter rule, where R_{ki} represented the i^{th} rule applicable to generating algorithm G_k. A parameter rule restricts the input parameter to G_k to some subset (might be improper) of the input domain I_k. The parameter rule thus restricts the generating algorithm to producing only some of the objects it is capable of generating.

The importance of the introduction of object process is that they provide the basis for the formation of objects. Let us associate with each object process $\mathbf{OP}_\eta$ the set of objects Θ_η that can be produced by $\mathbf{OP}_\eta$; we refer to these sets of potential objects as *classes*. Therefore for any object in the world, θ_j, there exists some class Θ_η such that $\theta_j \in \Theta_\eta$. To say that some world exhibits the property of natural modes is to place restrictions on the possible Θ_η that can exist in a given world. These restrictions arise either because of the generating algorithms available to produce objects, or because of the rules imposed on the generating algorithms, both of which can be viewed as determined by environmental pressures. We will also require that the number of classes present in the world be finite, agreeing with the intuition that there are a limited number of classes present and that the goal of the observer is to discover them.

Eventually, we will need to be able to express exactly the constraint of natural modes in terms of the classes produced by the object processes. We have already shown that a class will share the emergent properties of the object process. Therefore, a statement of the form "any two natural mode classes will differ by 'many' emergent properties" is an example (albeit imprecise) of defining natural modes in terms of the classes produced by the object processes.[8] A more specific class constraint might be ecologically based such as "symmetry is an emergent property." We will not pursue this discussion further in this section since our immediate goal is to identify the significant components of the visual classification problem and to understand how the concept of natural modes can be more formally expressed.

Finally, we assume that the observer views objects presented serially, in isolation. Thus the input to the observer is a *sequence*, written σ; we represent the first n objects in the sequence as σ_n. This notation is similar to that used for the presentation of sentences in formal learning theory [Osherson, etal.,1986]. We assume that the sequence is infinite, and that an infinite number of objects from each class is contained in the sequence. That is, the observer will never run out of data.

6.2 The Formal Observer

We now have a characterization of the world of objects which the observer is to classify. To complete the description of the classification problem we want to formalize the com-

[8] Note that although empirically this statement may be false, it's converse is always true: if two sets of objects can be shown to have different emergent properties then they must be long to different classes.

ponents contained in the head of the observer. From the POLY example we can identify several components, including a visual representation, the class constraints, the hypothesis (for classification) proposed by the observer, and the procedure used to propose the hypotheses. In this section we will build a complete model of an observer in which each of these components is explicitly represented. Our development of the observer will often parallel the development of a language learner presented in Formal Learning Theory.

In the POLY example each object was described using visual *features*, where a feature was some discrete, finite valued function defined for all objects. Examples of features included *"Length of Side"* and *"Containing Intersections"* as shown in tables 1–3. The description of the objects in terms of valued features was all the information available to the observer; these valued features, i.e. the properties, define the differentiable objects for the observer since any two (possibly different) objects which map into the same feature representation are identical to the observer. It is the description of the objects in terms of the valued features upon which the classification algorithm must operate.

To formalize the notion of visual description one must realize that although features are often selected as a method of representing objects for computation, all that is required is some form of visual representation. We define an object *representation* as a finite set Θ^* and a map $\mathcal{R}$ from the set all objects Θ to Θ^*, $\mathcal{R}: \Theta \longmapsto \Theta^*$. That is $\mathcal{R}$ is defined for all objects $\theta \in \Theta$. Although both the map $\mathcal{R}$ and the range of $\mathcal{R}$ comprise the representation, we will use the symbol $\mathcal{R}$ to stand for the representation, distinguishing between the mapping and the target set only when necessary. In POLY the features were the representation, mapping all the POLY objects onto into a finite set of ordered 6-tuples. The restriction of finiteness on the representation is chosen to agree with the intuition that there is some limit to the information encoded about an object by the observer. The major implication of this restriction is that although there may be an infinite number of objects (in fact uncountably infinite) there are only a finite number of *distinguishable* objects.

There is some difficulty in trying to decide what constitutes a visual representation because of the requirement that it be defined for all objects in the world. If we do not define the set of all objects, how can we say whether a proposed representation is actually acceptable, and can be computed on whatever object is presented? In fact, choosing a representation tacitly *defines* the set of objects, namely the domain of the representation. Having noted this point we will assume that either we are given a definition of "object", and have selected a representation $\mathcal{R}$ that spans the proper domain, or that we have chosen a representation $\mathcal{R}$ that was convenient (made explicit properties which we think will be useful to the classifier) and that we are content to be able to represent only the objects in the domain of $\mathcal{R}$.[9]

[9] Marr and Nishihara [1978] define the principle of scope relative to a representation. That principle is directly related to the idea that the selection of the representation defines the what can constitute an "object" for recognition.

Given a representation $\mathcal{R}$, we can now characterize the input that the observer actually sees, namely the representation of the sequence σ, notated as σ^*. (Any starred symbol will mean the element as expressed in the representation. Thus θ_j^* is the j^{th} object as represented by the observer.) σ^* is an ordered list $(\theta_1^*, \theta_2^*, \ldots)$; it is all the sensory information about the world provided to the observer.

In our POLY example, the next component of the classifier was the class constraints. These constraints restricted the classifications that could be proposed by the observer. However, to formally specify what constitutes a class constraint, we need to identify what is being proposed by the observer as objects are viewed. As such we define the observer's ***hypothesis***, $\mathcal{H}$, to be a set of sets of represented objects $\{\widehat{\Theta_1}^*, \widehat{\Theta_2}^*, \ldots, \widehat{\Theta_m}^*\}$. These objects are the classes proposed by the observer to match the actual (represented) classes $\{\Theta_1^*, \Theta_2^*, \ldots, \Theta_m^*\}$. Assuming the observer proposes some hypothesis after seeing each object in σ, we define $\mathcal{H}_n$ to be the hypothesis proposed after seeing the n^{th} object in σ. Notice that we have placed no constraint on whether every object seen in σ_n is contained in some class in $\mathcal{H}_n$, or whether objects not seen in σ_n are in $\mathcal{H}_n$. In fact, a very simple consequence of this formalism is that if only objects in σ^* are in $\mathcal{H}$, then every new object will change the hypothesis, preventing convergence until all (distinguishable) existing objects had been seen. In POLY, the hypothesis, $\mathcal{H}$, was the proposed groupings of objects with the associated defining features. That is, each class proposed was not only the objects seen so far that fit the feature description, but also the ***unseen*** objects which would satisfy that description. Therefore we were able to say that in the correctly constrained case the observer did converge to the proper classification.

In relation to our definition of hypothesis, the definition of the class constraints becomes clear. The class constraints provide an *evaluation* of a hypothesis in relation to the viewed objects (as described in the representation). As such the class constraints can be defined as an evaluating function $\mathcal{E}$, that takes as input the triplet of an initial sequence, a representation, and a hypothesis, and produces as output a natural number: $\mathcal{E}: \langle \sigma_n, \mathcal{R}, \mathcal{H}_n, \rangle \longmapsto N$. In our POLY examples, the class constraints evaluated to either a 1 or a 0 for any given hypothesis, where a one is interpreted as satisfying the constraints, and a 0 as not. The need for the class constraint evaluation function is easily appreciated by considering how the observer is supposed to know which hypothesis to propose given a sequence of objects. As discussed in the POLY example, without some external information (external to the sensory data) any arbitrary classification may be considered correct. The class constraint evaluation function provides that necessary external information by allowing the observer to evaluate the acceptability of a hypothesis. Therefore, how accurately the class constraints describe the relations between the real classes affects whether the observer has the ability to recover the natural classes.

Finally, we have the last component of the observer, the classification method, $\mathcal{M}$. The method is the procedure used by the observer to produce a hypothesis given each of the above components, $\langle \sigma, \mathcal{R}, \mathcal{E} \rangle$. Specifically, $\mathcal{M}$ produces some hypothesis $\mathcal{H}_n$ on

σ_n for all n. In the POLY example, the method used was an impoverished, incremental one, generating only hypotheses which were refinements of the previous proposed classes. A potential danger of such incremental methods is that for the same world of classes, different sequences will lead to the convergence of different hypotheses. Therefore, the method used by the observer to propose new hypotheses also affects whether or not the observer will have the competence to recover the actual classes. Just as with the constraint evaluation function, the method of proposing hypothesis needs to reflect the constraints operating in the world if the observer is going to be able to successfully categorize the objects in the world.

This completes our description of the observer. The complete observer can be a considered as a function, which given a representation $\mathcal{R}$, a constraint evaluation function $\mathcal{E}$, and a method $\mathcal{M}$, maps the set of initial sequences σ_n onto the set of hypotheses. That is after viewing each object object in the σ, the observer announces some hypothesis, $\mathcal{H}$.

6.3 Formal Problem Statement

Using the notation developed above let us restate the classification problem. In the ideal case, the observer would eventually propose a hypothesis which correctly describes the world and thereafter never deviate from it. Recall that the task of the observer is to recover the actual classes in the world $\{\Theta_\eta\}$. However, the observer only has knowledge about objects in the world expressed in terms of the representation. Once the representation is fixed, the best classification the observer could generate is one which matches the image of the real classes under the representation. To be able to state that the observer has correctly categorized the world, we must be able to relate the observers hypothesis expressed in terms of the representation to the actual classes present. In Appendix 1 we formally define a *class preserving* representation. Intuitively a representation is class preserving if the projection of the real classes under the representation preserves class membership. That is, disjoint classes in the world map to disjoint classes in Θ^*. Given this definition we now define a correct classification hypothesis: if there exists some class preserving representation which maps the world classes $\{\Theta_1, \Theta_2, \ldots, \Theta_m\}$ onto the hypothesized classes $\{\widehat{\Theta_1}^*, \widehat{\Theta_2}^*, \ldots, \widehat{\Theta_m}^*\}$ then the hypothesis is correct. Thus we have the following definition of successful classification:

> Consider a world of objects produced by a set of object processes $\{\mathbf{OP}_i\}$, defining the set of classes $\Theta_1, \Theta_2, \ldots, \Theta_m$. An observer, given a representation $\mathcal{R}$, a constraint evaluation function $\mathcal{E}$, and a method $\mathcal{M}$, is said to correctly classify the world of objects presented in some sequence σ if and only if there exists an n such that for all $m > n$, the hypotheses $\mathcal{H}_m = \mathcal{H}_n$ and $\mathcal{H}_n = \{\widehat{\Theta_1}^*, \widehat{\Theta_2}^*, \ldots, \widehat{\Theta_m}^*\}$ is the projection of the world classes $\{\Theta_1, \Theta_2, \ldots, \Theta_m\}$ under some class preserving representation.

Of course, stating that there exists *an* n simply says that in the limit the observer proposes the correct classification. In fact, as discussed by Osherson, Stob, and Weinstein [1986], other criteria might be desirable to include in defining a successful observer. For example it might be required that the observer be incrementally "better" where $\mathcal{H}_{n+1}$ is defined to be better than $\mathcal{H}_n$ if in some sense $\mathcal{H}_{n+1}$ is closer to the correct solution. Or some time or resource limitation may be imposed on the observer, making the best observer one that can generate the closest classification in some fixed amount of time or computational space. We will consider some of these issues when discussing observers suited to the natural world.

7 Traditional Cluster Analysis

The purpose of this section is to consider some of the current methodologies for performing the object categorization task in light of the formulation presented above. Specifically, the methods of *cluster analysis* might appear to be sufficient to address the issues raised. We argue that this is not so for several reasons, some of which are in fact fundamental to cluster analysis.

7.1 Standard Clustering Techniques

Most classical methods for doing cluster analysis can be described by the following outline [Duda and Hart, 1973][10] :

1) Measure some feature vector for each point in a sample.
2) Transform the data according to some assumption about the statistics of the features.
3) For some number c, find c clusters of the sample points which satisfy a clustering criterion provided by the programmer.

Step 2 is usually some form of normalization of the data, with a favorite technique being the scaling of each feature to yield normal distributions for each dimension. This normalization is often required because the criteria in step 3 is usually some form of distance metric that is sensitive to the absolute scale of each dimension. Notice that step 3 does not require finding the *best* c groups, rather just good clusters. Such sub-optimal solutions are usually considered adequate because of the computationally intractable

[10] We should mention that Michalski has advocated a different approach to cluster analysis than is usually considered. See [Michalski and Stepp, 1983] for discussions.

problem of finding the optimal clustering. Therefore, iterative, locally optimizing schemes are ordinarily employed.

Before considering the more fundamental inadequacies of cluster analysis to solve the problem of visual categorization as presented here, we note that the algorithms by which cluster analysis theory are implemented are *not* well suited to object classification.

For example, most cluster analysis programs (though see Michalski and Stepp, [1983]) use a simple basis set of features for describing the data. This presupposes that there is some *one* set of features which provides the important information for performing categorization. Furthermore, some distance metric is then computed on this set, with the metric remaining uniform throughout the feature space. We believe that this approach to object recognition (categorization) is ill-motivated. Given that different object processes constrain quite different properties of objects, one would suspect that the "important" properties would vary from class to class. Thus, although one large set of features might be used to describe the objects, different classes would constrain different features, making uniform distance metrics inappropriate.

Leaving the algorithmic issues aside, we will consider two principal components of the object classification problem which are not addressed by cluster analysis theory. They are the concepts of an objectively correct classification, and the need for a performance/competence distinction.

7.2 Correct versus Desired Categories

The most important component of the object classification problem which is missing from cluster analysis is the notion of a "correct" classification. In our description of the classification problem, the entire motivation is provided by the existence of natural, objective categories — natural modes. That is, there exist actual classes for the observer to recover, as defined by the constraints operating in the environment. The goal, then, in designing an observer, is to understand the constraints operating on object processes which give rise to the Natural Modes and to embed that knowledge in the classification procedure itself, permitting the recovery of the *real* classes. In cluster analysis, the criteria for clustering are based solely on the desired form of a cluster, usually with respect to some further operations to be performed on the groups found. That is, the only measure of success of a clustering is whether the clusters "look good." Without the addition of the objective classes, the goal of a correct clustering is undefined.

In fairness to proponents of cluster analysis, often cluster analysis is used specifically because the constraints operating on the domain are unknown, and the goal is that the clusters found should provide some insight into the underlying structure present. For example, consider the problem of trying to cluster a population of micro-organisms in a sample of pond water. Let us assume that nothing is known about DNA or the micro-

biology that is the actual underlying cause for the differences between the specimens. Thus in hoping to learn something about the underlying structure one might attempt to find clusters as defined by some measured data and some preconceived notion of what a cluster should look like. But, under what conditions would such a clustering technique be successful? Only if the properties measured are properties constrained in some systematic way by the underlying structure (micro-biology) which agreed with the preconceived criterion for good clustering. That is, discovery of structure is only to be expected when one is actually measuring something which is constrained by some underlying process, and when that constraint acts in a way which is consistent with the types of structure which are assumed to be important. Therefore the real work to be accomplished in finding structure present in a set is to understand the domain well enough to be able to predict what types of structure might be present.[11] This is our idea of building into the observer the constraints of the Natural Modes, for then we can specify such things as class constraints and examine under what conditions the classification procedure will converge to the correct classes. This critical point of understanding the domain, and using that knowledge to find the right classes, is not addressed by cluster analysis.

7.3 Competence versus Performance

We have been using the term *competence* in a manner slightly different than that which is usually found in discussions about the knowledge and abilities of some problem solving agent. The normal distinction between competence and *performance* in a domain is perhaps best articulated by Chomsky [1965] and, in relation to Artificial Intelligence, by Winston [1979, 1984] and Marr [1977]. Performance reflects the details of how a procedure utilizes some knowledge in accomplishing a task, while competence is the knowledge itself. Normally the competence cannot be measured directly, but must be inferred from the performance. The traditional example considered is that of the natural language grammar maintained by an individual. The grammar represents the knowledge (the competence) acquired by the individual; how well he can parse strings of words is the performance from which the grammar is inferred.

For the discussions presented here, we make the following distinction between competence and performance. We regard performance as the demonstrated ability of a problem solving agent to solve a particular problem. Using another linguistic example, a child learning English has demonstrated the ability to learn English in the particular environ-

[11] An example of an attempt to discover structure in data which highlights well the problem of only being able to discover what you are looking for is Langley's work on *Bacon*. That program, designed to discover lawful relations, works well in some domains (e.g. chemistry) because the types of relations that are considered (simple arithmetic operations) are in fact the correct descriptions for the domains. [Langley, Bradshaw, and Simon 1983]

ment in which the learning transpired; this is an example of performance. We define competence, however, to be the set of *potential* problems that could be solved. Continuing with the linguistic example, it is currently believed that normal children have the competence to learn any natural language given any normal speaking environment.[12] This statement of competence is a conjecture, and cannot currently be proven for two reasons. First, we do not as yet have a sufficient understanding of what natural languages are to be able to characterize them formally. Second, we cannot peer inside the head of the child and read out the algorithm being used to compute the grammar of the language. Therefore, we must assert competence from several examples of performance.

However, when designing a problem solving agent for a particular task one can in principle consider the competence of the system. In relation to our problem of visual categorization, one would like to be able to characterize both the world and the observer such that one could demonstrate competence on the part of the observer to recover the classes present in the world. Consider the following example of a "nativist" observer. The method used by this observer is to hypothesize one particular hypothesis $\mathcal{H}^0$, *regardless of the objects viewed!*[13] Although such an observer may seem rather impoverished, it is certainly a legitimate classifier. Furthermore, this observer can be said to have the competence to recover the classes in any world which is correctly classified by $\mathcal{H}^0$. For any σ drawn from such a world, the nativist observer would recover the true classes.

The performance/competence distinction is not usually addressed in cluster analysis (though see Jardine and Sibson [1971]). Most procedures are proposed algorithms for finding clusters, and there competence is asserted by demonstrating performance (both successes and failures) on particular tasks. Certainly, this lack of competence analysis can be attributed to the complexity of the algorithms, which are often large iterative searches whose successes or failures are determined by many, sometimes mysterious, properties of the data. Because of this lack of analysis, it is difficult to understand under what conditions the algorithm will succeed, and when it will fail. Because one of the primary goals in understanding object categorization is to understand what information is necessary to make an observer competent to recover the classes in a given world, cluster analysis does not provide much insight into this question.

---

[12] One problem with this analogy is that natural language is sometimes defined as any language that can be learned by a normal child, making the statement of competence a tautology. If however we assume that there exists a set of criteria (as yet unknown) which is independent of the abilities of a learner and which defines the set of natural languages, then the competence statement is non-trivial.

[13] We use the super-script notation $\mathcal{H}^i$ to represent some particular hypothesis. This is in contrast to the sub-script notation which represents the place in the σ when the hypothesis was proposed. Thus, the equation $\mathcal{H}_n = \mathcal{H}^i$ indicates that the hypothesis proposed after viewing the n^{th} object was the particular hypothesis $\mathcal{H}^i$.

8 The Natural World

Having considered the classification problem in both a toy world (POLY) and in the abstract, we now consider the real, natural world. Several key issues relative to our discussion about visual classification are immediately apparent, including how well the model of objects and natural modes describes the real world, and how the natural modes in the world are constrained. Also, the resources and goals of the real observers — human beings — are a factor in considering a visual classification system. We will address each of this issues briefly. For the most part, we leave some important questions about the exact nature of the real world unanswered; our goal here is to lay out some of the interesting problems to be solved.

8.1 Natural Object Processes and Real Natural Modes

The first question to be addressed is the degree to which the world is really clustered in to natural classes. More concisely, is it true that "there exist *objective* categories of objects in the world?"

Clearly, the answer lies somewhere between a definite yes and a definite no. Certainly, for some tasks, the differences between one's own German Shepherd and someone else's may be critical, making the idea that there is some well defined set of classes seem untrue. But as a crude first categorical statement about the nature of the world, there do indeed seem to be clearly defined classes. For example, consider the biological kingdom. Of all the possible species that can be created, of all the possible DNA codes which could produce an organism, there are only a relative few that exist [Stebbins and Ayala, 1985]. In fact, if one examines one aspect of DNA coding, the complexity of the code itself, one will observe a clumping in the distribution, where the divisions between the groups lie along boundaries which agree with other taxonomic divisions. This clumping is presumably caused by the interaction between the organisms ("objects") produced by the DNA and the environment causing only some forms to propagate.

But what about our model of an object process — the pair of a generating algorithm plus some restrictive parameter rule? As mentioned in section 3, it is possible to have a completely general generating procedure (a.k.a. Universal Turing Machine) which can produce any object, depending upon the parameter chosen. Therefore our distinction of emergent versus parametric property has been artificially created. We do this however to agree with the intuition that Nature has constructed relatively few methods for producing objects, and that it is the constraints placed on those methods which produce particular types of objects. In fact, much current vision research is focused on understanding the physical processes which produce objects in the natural world, to permit the creation of models which capture the structure imposed on objects by these processes [Pentland,

1985; Kass and Witkin, 1985]. As such, we believe that separately representing the generating algorithm and its constraints will be to our advantage when trying to model the natural modes as actually present in our world.

Finally, we note the possibility of there existing a hierarchy of natural modes in the world and the ability of our classification system to accommodate such a structure. For example, consider again the class of dogs and within that class the subclasses of German Shepherds and Siberian Huskies. To say that one particular level is *the* level of natural modes may be an over-simplification. One could imagine a tree-like classification system where one type of evaluation function may converge to the class of dogs, after which a more refined set of evaluation functions would distinguish between types of dogs. In fact, one strategy to simplify the control problem of generating hypotheses (see section 6.2) is to achieve a hierarchical description, with a new classifier attempting to find sub-classes within only one of the classes of the previous level. From our POLY example the class of D,H,J may be viewed in this way as a subclass of B,D,F,G,H,J.

8.2 Constraints on the Observer and Criteria for Success

In defining the formal problem of classification we define an abstract notion of successful classification, namely eventually being able to obtain the correct classes. The important point there was that we were addressing the issue of competence; the particular observer has the *ability* to recover correct classes. In the real world however. additional considerations become important. For example, it would be desirable for the observer to converge rapidly to a good approximation to the correct classes, increasing his chances that at any given moment he has a sufficiently correct categorization to make important inferences. Although this may restrict the types of classes he can discover,[14] he may be willing to sacrifice this power for a wrong-but-close inference. Alternatively, the observer may require a slowly varying hypothesis, where no one object causes a radical change in the current hypothesis. Expressed in our notation we might restrict $\mathcal{H}_{n+1}$ to differ from $\mathcal{H}_n$ by no more than 2 (or more generally k) classes. These different constraints will restrict the type of world classes that can be recovered by the observer.

Additionally, we may have constraints placed on the observer in terms of computing resource and time. These constraints will affect the methods available to the observer to make hypotheses. For example recall that one naive method discussed in the POLY example was to consider all possible partitions of the data, and to choose the one that satisfied the constraints best (assume that the constraint function is multi-valued and that it is not simply a matter of finding a partition which satisfies a constraint). The combinatorics of such a method exploded so quickly that any resource limit would be violated early in the classification process. As such, in the real world, with millions of

[14] See Osherson, etal. [1986] for the elegant demonstration of a learner who, because he never hypothesizes a "less correct" theory, cannot learn a particular collection of languages.

objects (and thousands of classes) such a method becomes impractical given any reasonable finite amount of time. However, incremental methods of producing classifications lack the power of global procedures, and face problems similar to those encountered by hill-climbing problem-solvers. As we continue to develop our model of the observer, we will need to be able to express these types of constraints and trade-offs easily within our notation.

8.3 An Oracle

The premise of the classification system described here is that the observer is provided with no information about how accurate its classification is except as embedded in the evaluation function. In the natural world however, there are alternative sources of information. A simple case is being told that a categorization judgment is incorrect by some authoritative outside source: someone corrects your categorizing a rabbit as a fox. A more subtle source of information is an experiment performed by the observer in the course of his interaction with the environment. For example a prediction is made about the behavior of an object based upon it's categorization and *upon the assumption that objects that categorized similarly will behave similarly.* If the experiment fails (e.g. the tiger which he thought was the same as a monkey chases him instead of running away), then there is some new information added to the system. Using our model, this represents a modification of the evaluation function. That is, the evaluation function becomes better matched to the world as more information is added. An interesting conjecture is that the initial evaluation function is capable of only achieving a crude classification of objects in the world, and that the addition of new information is required to develop more powerful and appropriate class evaluation functions.

9 Summary

How it is possible for an observer to categorize objects in the world such that the classes generated are useful for inferring important properties about the objects? Such a categorization would be the foundation for general object recognition. We have drawn upon the work of Osherson, Stob, and Weinstein [1986] in developing a formal description of the object classification problem and have identified two necessary conditions for successful object classification using vision or other sensory data.

First, the objects themselves must exhibit structure such that unobservable properties can be inferred from observable ones. We propose that this structure obey the

"Principle of Natural Modes" which requires that Nature has a limited number of methods for producing objects that will survive under environmental pressure. As a model of object formation we introduce *object processes* consisting of a generating algorithm capable of producing objects, plus some parameter rule that restricts which objects the algorithm will create. We associate with each object process the set of objects it can produce; that set defines a *class*. The goal of the observer is to recover these classes.

Second, to discover the classes present in the world, the observer must have a classification system which correctly matches the structure exhibited in the world. The system we provide the observer is described as having three components: 1) a *representation* — the data structure used to describe the objects; 2) an *evaluation* function — a function required to select among alternative classifications; 3) a *method* — the procedure used to generate a *hypothesis* given a *sequence* of objects. Each of these components must be suitably matched to the structure in the world or the observer will not be able to correctly categorize the objects. Using these components we have defined *successful classification*, allowing us to consider the competence of an observer to categorize a particular world.

References

Abelson, H. and A. diSessa [1981], *Turtle Geometry*, MIT Press, Cambridge, MA.

Anderburg, M. R. [1973], *Cluster Analysis for Applications*, Academic Press, New York.

Chomsky, N. [1965], *Aspects of the Theory of Syntax*, MIT Press, Cambridge, MA

Duda, R. and P. Hart [1973], *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York.

Gibson, J. J. [1979], *The Ecological Approach to Visual Perception*, Houghton Mifflin Co., Boston.

Jardine, N. and R. Sibson [1971], *Mathematical Taxonomy*, John Wiley & Sons, London.

Jolicoeur, P., M. A. Gluck, and S .M. Kosslyn,"Pictures and Names: Making the Connection," *Cognitive Psychology*, vol. 16, No. 2, pp. 243–275.

Kass, M. and A. P. Witkin [1985], "Analyzing Oriented Patterns," Schlumberger AI TR-42. Also presented at IJCAI, 1985.

Lance, G. N. and W. T. Williams [1967], "A general theory of classificatory sorting strategies. II. Clustering Systems", *Computer J.*, vol. 10, No. 3, pp.271–277.

Langley, P., G. L. Bradshaw and H. A. Simon [1983], "Rediscovering Chemistry With the BACON System," in *Machine Learning — An Artificial Intelligence Approach,* ed. by R. Michalski, J. Carbonell, and T. Mitchell, Tioga Publishing Co., Palo Alto, CA.

Levi, B. [1986], "New Global Formalism Describe Paths to Turbulence," *Physics Today,* vol. 39, No. 4, pp.17–18.

Lozano-Perez, T. [1985], "Shape-from-Function", MIT Artificial Intelligence Lab. Revolving Seminar.

Marr, D. [1977], "Artificial Intelligence — A Personal View," *Artificial Intelligence,* vol. 9, No ?, pp. 37-48.

Marr, D. [1970], "A theory of cerebral neocortex", *Proc. R. Soc. Lond. B,* vol. 200, pp. 269-294.

Marr, D. and H. K. Nishihara [1978], "Representation and recognition of the spatial organization of three-dimensional shapes," *Proc. R. Soc. Lond. B,* vol. 200, pp. 269-294.

Mayr, E. [1984], "Species Concepts and Their Applications" in *Conceptual Issues in Evolutionary Biology: An Anthology*, ed. E. Sober, MIT Press, Cambridge, pp. 531-541.

McMahon, T. A. [1975], "Using body size to understand the structural design of animals: Quadruped Locomotion," *J. Applied Physiology,* vol. 39, pp. 619-627.

Michalski, R. S. and R. E. Stepp [1983], "Learning From Observation: Conceptual Clustering," in *Machine Learning — An Artificial Intelligence Approach,* ed. by R. Michalski, J. Carbonell, and T. Mitchell, Tioga Publishing Co., Palo Alto, CA.

Osherson, D., M. Stob, and S. Weinstein [1986], *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists*, MIT Press, Cambridge, MA.

Pentland, A. [1985], "Perceptual Organization and the Representation of Natural Form," SRI Technical Note 357.

Rosch, E. [1978], "Principle of Categorization," in *Cognition and Categorization,* ed. by E. Rosch and B. Lloyd, Lawrence Erlbaum Associates, Hillsdale, New Jersey, pp. 28-49.

Rota, G. C. [1964], "The number of partitions," *American Math. Mon.,* vol. 71, pp. 498-504.

Stebbins, G. L.. and F.J. Ayala [1985], "Evolution of Darwinsim", *Scientific American,* Vol. 253, No. 1, pp. 74.

Thompson, D. [1961] *On Growth and Form*, Cambridge University Press, Cambridge, Great Britain.

Winston, P. H. [1979], "Learning and Reasoning by Analogy", MIT Artificial Intelligence Lab. Memo 520. Also appears in abbreviated form in *Comm. ACM*, vol. 23, No. 12, 1980.

Winston, P. H. [1985], *Artificial Intelligence, Second Edition*, Addison-Wesley, Reading, MA.

Winston, P. H., T. O. Binford, B. Katz, and M. Lowry [1983], "Learning Physical Descriptions from Functional Definitions, Examples, and Precedents," MIT Artificial

Intelligence Lab. Memo 679.

Witkin, A. P. and J. M. Tenenbaum [1983], "On the Role of Structure in Vision" in *Human and Machine Vision* ed. J. Beck, B. Hope, and A. Rosenfeld, Academic Press, New York.

Appendix 1

Here we define a *class preserving* representation. We assume the set of all possible objects is given; it may be uncountable. We also assume there exist some set of classes, whose union is a subset of the set of all possible objects. If one is uncomfortable with the notion of the set of all possible objects, then first choose the representation and then let the domain of the representation be the set of possible objects.

Given the set of possible objects Θ, choose some finite set Φ and a total-function $\mathcal{R}$ such that $\mathcal{R}: \Theta \longmapsto \Phi$. We refer to the total function $\mathcal{R}$ (and its associated range) as a *representation.* Given $\theta \in \Theta$, define θ^* such that $\theta \overset{\mathcal{R}}{\mapsto} \theta^*$. We write $\mathcal{R}(\theta) = \theta^*$. Let ϕ be some arbitrary member of Φ. The equation $\phi = \theta_i^*$ means that ϕ is equal to $\mathcal{R}(\theta_i)$ and we say that ϕ is the *image* of θ_i. Thus, we define $\Theta^* \subseteq \Phi$ to be the set of all elements of Φ which are images, and we can write $\mathcal{R}: \Theta \longmapsto \Theta^*$.

Now consider a set of classes $\{\Theta_\eta\}, \eta = 1, \ldots, m$ such that $\cup_\eta \Theta_\eta \subseteq \Theta$. We define the *projection* of the set of classes under the representation $\mathcal{R}$ as the set $\{\Theta_\eta^*\}, \eta = 1, \ldots, m$, with each Θ_η^* defined as:

$$\Theta_\eta^* = \{\phi \in \Phi | \; \exists \, \theta_i \in \Theta_\eta \text{ such that } \mathcal{R}(\theta_i) = \phi\}$$

That is, Θ_η^* is the set of images of the objects in Θ_η. Therefore if $\theta_i \in \Theta_\eta$ then $\theta_i^* \in \Theta_\eta^*$.

We can now define a class preserving representation:

> *$\mathcal{R}$ is class preserving if and only if: if $\phi \in \Phi$ is an element of Θ_η^*, then there does not exist a θ_i, $\theta_i \notin \Theta_\eta$, $\theta_i \in \Theta_\mu$, such that $\mathcal{R}(\theta_i) = \phi$.*

That is, if some ϕ is the image of some object in Θ_η, then there is no object which is *not* in Θ_η but is in some other Θ_μ, which is also mapped to ϕ by $\mathcal{R}$.

Glossary

G_k — The k^{th} *generating algorithm*, which produces objects when given an input parameter.

I_k — The input domain to the k^{th} generating algorithm.

$\mathcal{A}$ — An input parameter to a generating algorithm.

R_{ki} — The i^{th} *parameter rule* for the k^{th} generating algorithm which restricts the input to G_k to be some subset of I_k.

$\langle G_k, R_{ki} \rangle$ — The *object process* which consists of generating algorithm G_k and parameter rule R_{ki}.

$\mathbf{OP}_\eta$ — A particular object process indexed by η.

$\{\theta\}_\eta$ — The set of objects produced by the object process $\mathbf{OP}_\eta$, called a *class*.

Θ — The set of all possible objects.

Θ_η — The set of objects which comprise the class η. Same as $\{\theta\}_\eta$.

Θ_W — The set of all objects present in the world, called the *population*. $\Theta_W \subseteq \Theta_1 \cup \Theta_2 \cup \ldots \cup \Theta_m$.

σ — The sequence of objects viewed by an observer.

σ_n — The initial part of the sequence containing the first n objects viewed.

Θ^* — A finite set used to represent objects by mapping Θ onto Θ^*. (See appendix 1 for a formal definition.)

$\mathcal{R}$ — The mapping from Θ to Θ^*. The symbol $\mathcal{R}$ stands for the *representation* which includes both Θ^* and the mapping.

σ^* — The sequence of objects as represented by the observer. (Any starred object/set is the representation of that object/set.)

$\widehat{\Theta_\eta}^*$ — A set of represented objects proposed by the observer to be a class.

$\mathcal{H}$ — A *hypothesis* proposed by the observer to describe the classes present in the world. $\mathcal{H} = \{\widehat{\Theta_1}^*, \widehat{\Theta_2}^*, \ldots, \widehat{\Theta_m}^*\}$,

$\mathcal{H}_n$ — The hypothesis proposed after viewing the n^{th} object.

$\mathcal{E}$ — An *evaluation* function which evaluates a hypothesis given a representation and the sequence of objects viewed. $\mathcal{E}: \langle \sigma_n, \mathcal{R}, \mathcal{H}_n \rangle \mapsto N$.

$\mathcal{M}$ A *method* for producing hypothesis after each initial sequence. Given the triplet $\langle \sigma_n, \mathcal{R}, \mathcal{E} \rangle$ the method produces a hypothesis $\mathcal{H}_n$.

$\mathcal{H}^i$ Some particular hypothesis.